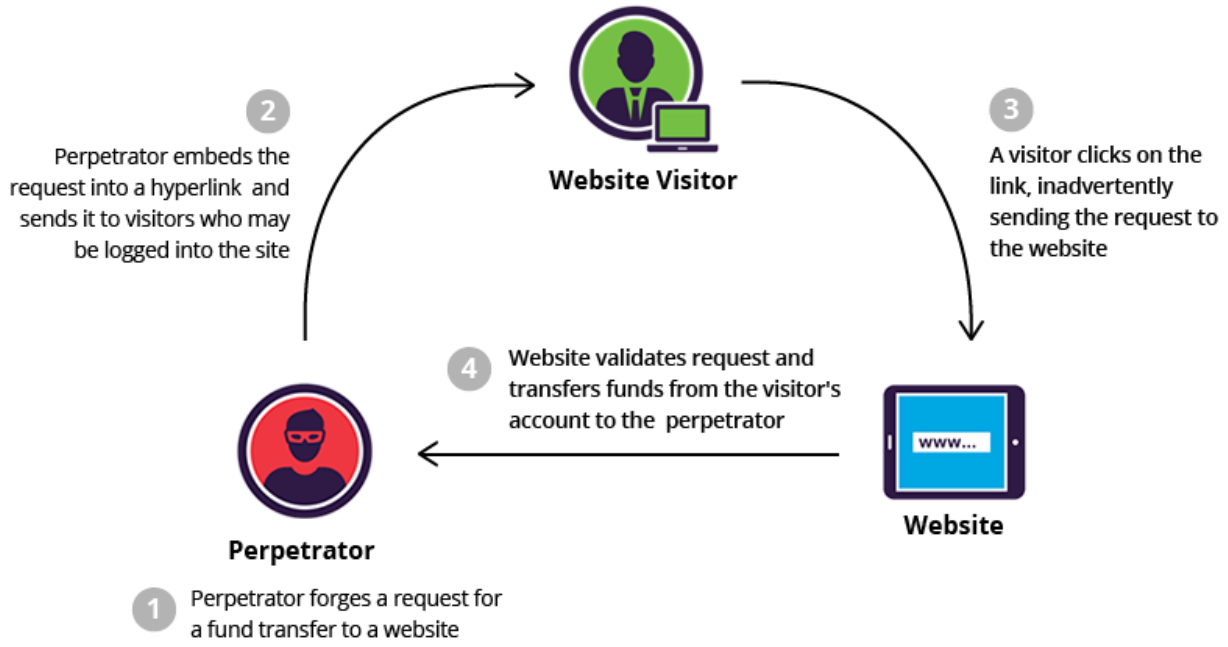
Cross Site Request Forgery - CSRF

What is CSRF?
CSRF is an attack in which the attacker tricks the victim's browser to make an unwanted request to the application to which he is already logged in. If allows an attacker to run the actions performed by an authenticated user without knowing his password.



How does a CSRF attack work?
Let's say that the online banking application is built using the GET method to submit a transfer request. As such, Bob's request to transfer $500 to Alice (with account number 213367) might look like this:

GET https://samplebank.com/onlinebanking/transfer?amount=500&accountNumber=213367 HTTP/1.1

Aligning with the first requirement to successfully launch a CSRF attack, an attacker must craft a malicious URL to transfer $5,000 to the account 425654:

https://samplebank.com/onlinebanking/transfer?amount=5000&accountNumber=425654

An attacker can trick Bob into loading the malicious URL The following is an example of a disguised URL:

<img src =
“https://samplebank.com/onlinebanking/transfer?amount=5000&accountNumber=425654” width=“0” height= “0”>

How do you test for CSRF vulnerability?

1. **Manual :** Go engagement tools on burp and click “Create CSRF POC”. You’ll get a HTML page. Make sure you test that page, whether CSRF is possible or not with another account.

2. **Automated:**
   CSRF Tester:
   CSRF Tester is a project by OWASP, created by a group of developers for developers, to verify the integrity of HTTP requests in their web applications. CSRF Tester provides a PHP library and an Apache Module for cautious mitigation.

What is the impact?

- Changing the victim’s email address or password.
- Purchasing anything.
- Making a bank transaction.
- Explicitly logging out the user from his account.

Mitigations

- Implement “Anti-CSRF tokens”
- Use of Same-Site cookies attributes for session cookies, which can only be sent if the request is being made from the origin related to the cookie.
- Do not use GET requests for state-changing operations.
- Identifying Source Origin (via Origin/Referer header)
- One-time Token should be implemented for the defense of User Interaction based CSRF.

References

https://www.synopsys.com/glossary/what-is-csrf.html#:~:text=Cross%2DSite%20Request%20Forgery%20(CSRF)%20is%20an%20attack%20that,has%20in%20an%20authenticated%20user

https://asfiyashaikh.medium.com/cross-site-request-forgery-csrf-8ce6f9ee0379

https://www.codecademy.com/learn/seasp-defending-node-applications-from-sql-injection-xss-csrf-attacks/modules/seasp-preventing-cross-site-request-forgery-csrf-attacks/cheatsheet

https://www.imperva.com/learn/application-security/csrf-cross-site-request-forgery/